CLAIMS

We Claim:

1 A method for providing data integration and exchange between a plurality of client applications over a network, wherein each of the client applications access a respective data source, and wherein the data sources of each of the client applications may be stored in different formats and not directly accessible by the other client applications, the method comprising the steps of:

(a) providing an adapter API that provides a first set of methods for the client applications to use to translate data into XML; and

(b) modifying each of the client applications to invoke the methods in the adapter API to convert data in the respective data sources into XML format and to have the XML formatted data imported into a database on a server, thereby standardizing the data from the data sources.

2 The method of claim 1 further including step of:

(c) including a second set of methods in the adapter API for the client applications to use that export data from the database into the client applications using standard Web services.

3 The method of claim 2 further including the step of: registering the respective data sources with a schema registry in order to create a schema definition and a document type definition (DTD).

4 The method of claim 3 wherein step (b) further includes the step of: using a respective DTD to validate the XML formatted data generated from the corresponding data source prior to importing the XML formatted data into the database.

5 The method of claim 3 further including the step of: registering the data sources with the schema registry by accessing a server web site with a Web browser and entering information regarding the schema in fields of a table.

6 The method of claim 3 wherein the adapter API includes an XML API comprising the first and second methods, wherein the first set of methods comprises a Writer API, and the second set comprises a Reader API.

7 The method of claim 6 wherein the client applications are modified with generator logic that makes calls to methods comprising the adapter API, wherein once called, the Writer API converts data into XML format in memory and saves the XML format data in an XML file, which is then transported to the server.

8 The method of claim 6 wherein the adapter further includes a standardized API comprising SOAP, TCP/IP, and HTTP libraries.

9 The method of claim 8 wherein the adapter further includes verification code that verifies the generated XML data against the DTD defined in schema registry.

10 The method of claim 6 wherein the adapter API is published using web services description language (WSDL).

11 The method of claim 7 wherein the server includes an import repository for receiving the XML files generated by the client applications.

12 The method of claim 11 wherein the server includes an XML loader that parses each of the XML files in the import repository and stores name/value pairs in the database according to a data hierarchy of the corresponding data source.

13 A computer-readable medium containing program instructions for providing data integration and exchange between a plurality of client applications over a network, wherein each of the client applications access a respective data source, and wherein the data sources of each of the client applications may be stored in different formats and not directly accessible by the other client applications, the program instructions for:

(a) providing an adapter API that provides a first set of methods for the client applications to use to translate data into XML; and

(b) modifying each of the client applications to invoke the methods in the adapter API to convert data in the respective data sources into XML format and to have the XML formatted data imported into a database on a server, thereby standardizing the data from the data sources.

14 The computer-readable medium of claim 13 further including instruction of:

(c) including a second set of methods in the adapter API for the client applications to use that export data from the database into the client applications using standard Web services.

15 The computer-readable medium of claim 14 further including the instruction of: registering the respective data sources with a schema registry in order to create a schema definition and a document type definition (DTD).

16 The computer-readable medium of claim 15 wherein instruction (b) further includes the instruction of: using a respective DTD to validate the XML formatted data generated from the corresponding data source prior to importing the XML formatted data into the database.

17 The computer-readable medium of claim 15 further including the instruction of: registering the data sources with the schema registry by accessing a server web site with a Web browser and entering information regarding the schema in fields of a table.

18 The computer-readable medium of claim 15 wherein the adapter API includes an XML API comprising the first and second methods, wherein the first set of methods comprises a Writer API, and the second set comprises a Reader API.

19 The computer-readable medium of claim 18 wherein the client applications are modified with generator logic that makes calls to methods comprising the adapter API, wherein once called, the Writer API converts data into XML format in memory and saves the XML format data in an XML file, which is transported to the server.

20 The computer-readable medium of claim 18 wherein the adapter further includes a standardized API comprising SOAP, TCP/IP, and HTTP libraries.

21 The computer-readable medium of claim 20 wherein the adapter further includes verification code that verifies the generated XML data against the DTD defined in schema registry.

22 The computer-readable medium of claim 18 wherein the adapter API is published using web services description language (WSDL).

23 The computer-readable medium of claim 19 wherein the server includes an import repository for receiving the XML files generated by the client applications.

24 The computer-readable medium of claim 23 wherein the server includes an XML loader that parses each of the XML files in the import repository and stores name/value pairs in the database according to a data hierarchy of the corresponding data source.

25 A data integration system comprising:

a network;

a server coupled to the network, the server including a schema registry, a database, and a published adapter API that provides a first set of methods for translating data into XML; and

a plurality of client applications coupled to the network and in communication with the server, wherein each of the client applications access a respective data source, and wherein the data sources of each of the client applications may be stored in different formats and are not directly accessible by the other client applications, and

wherein at least a portion of the client applications includes a corresponding schema definition and document type document (DTD) registered with the schema registry, and the portion of the client applications includes generation logic for making calls to the methods in the adapter API, such that data in the respective data sources are converted into XML format and transferred to the server, wherein the XML data is verified against the corresponding DTD prior to storage in the database, thereby standardizing the data from the data sources.

26 The system of claim 25 wherein the adapter API further includes a second set of methods for the client applications to invoke that export data from the database into the client applications using standard Web services.

27 The system of claim 26 wherein the adapter API includes an XML API comprising the first and second methods, wherein the first set of methods comprises a Writer API and the second set comprises a Reader API.

28 The system of claim 27 wherein the server further includes a schema generator for generating the schema definition, a DTD generator for generating the DTD, and an adapter software kit that is downloaded from the server and used to incorporate the adapter API into the client applications.

29 The system of claim 28 wherein the server further includes an import repository for receiving XML files generated by the client applications.

30 The system of claim 29 wherein the server includes an XML loader that parses each of the XML files in the import repository and stores name/value pairs in the database according to a data hierarchy of the corresponding data source.